Japanese Patent No. 2000-228685

Job No.: 0-00958-1
Ref.: JP2000-228685 AND JP2001-285205//PU020488 JP/MAP(DAVIDA)/ORDER NO. ART338
Translated from Japanese by the McElroy Translation Company
800-531-9977 customerservice@mcelroytranslation.com

(19) JAPANESE PATENT OFFICE (JP) (12) KOKAI TOKUHYO PATENT GAZETTE (A) (11) PATENT APPLICATION PUBLICATION NO. 2000-228685
(43) Publication Date: August 15, 2000

(51) Int. Cl.[7]:	Identification Codes:	Theme Codes (Reference):
H 04 L 29/06		5C064
H 04 N 7/20	630	5K034
//H 04 H 1/00		
H 04 N 7/167		
H 04 L 13/00	305B	
H 04 N 7/20	630	
H 04 H 1/00	Z	
H 04 N 7/167	Z	

Examination Request: Not filed No. of Claims: 13 OL (Total of 12 pages)

(21) Filing No.: Hei 11[1999]-29100

(22) Filing Date: February 5, 1999

(71) Applicant: 000002185
Sony Corporation
6-7-35 Kitashinagwa
Shinagawa-ku, Tokyo

(72) Inventor: Noboru Fujii
Sony Corporation
6-7-35 Kitashinagwa
Shinagawa-ku, Tokyo

(74) Agent: 100067736
Akira Koike, Patent Attorney (and two others)

F Terms (Reference):
5C064 CA14 CC02 DA02 DA09
5K034 AA19 AA20 EE03 EE10 EE11
FF01 HH02 HH12 HH16 HH61
JJ11 JJ21 MM25

[There are no amendments to this patent.]

(54) [Title] INTERFACE DEVICE AND RECEIVER

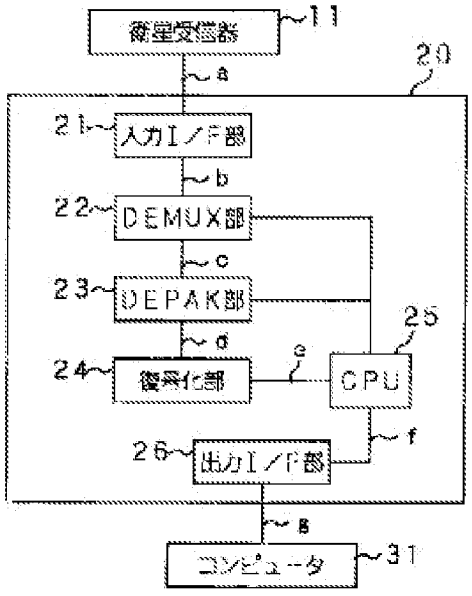
(57) Abstract

Problem

To implement data processing on a personal computer utilizing satellite communication and satellite broadcasting.

Means to solve

[A receiver] that has a satellite receiver 11 that receives communications from a satellite, a demultiplexer 22 that separates a prescribed channel from the multiple-channel data streams; a depacketizer 24 [sic; 23] that reconstructs data; a decoding unit 24 that decodes encoded data; and a CPU 25 that exchanges data in a prescribed format; in addition, [the receiver] converts a data stream supplied from a satellite to [the format of] an external interface of a personal computer 31.



Key:	11	Satellite receiver
	21	Input I/F unit
	22	Demux unit
	23	Depak unit
	24	Decoding unit
	26	Output I/F unit
	31	Computer

Claims

1. Interface device characterized by: a first interface means to which is input a data stream received as a communication from a satellite;

a demultiplexer that separates the packets for a desired channel from the multiple-channel data streams supplied from the aforementioned first interface means;

a depacketizer that reconstructs data from the packets separated by the aforementioned demultiplexer;

a control means that converts the data reconstructed by the aforementioned depacketizer in a prescribed format;

and a second interface means that is connected to an external computer.

2. Interface device according to Claim 1, characterized in that the aforementioned control means is controlled by the external computer connected to the aforementioned second interface means.

3. Interface device according to Claim 2, characterized in that: a data stream is input to the aforementioned first interface means from a receiver that receives data transmitted from a satellite;

the aforementioned first interface means has a function whereby it transmits a control signal to the aforementioned receiver;

and the external computer connected to the aforementioned second interface means controls the aforementioned receiver through the aforementioned control means and the aforementioned first interface means.

4. Interface device according to Claim 1, characterized in that the data that forms the aforementioned packets is encoded,

and [characterized in that] it is further equipped with a decoding means that decodes the encoded data that has been reconstructed by the aforementioned depacketizer.

5. Interface device according to Claim 1, characterized in that the aforementioned control means is equipped with a correspondence table for Internet protocol addresses and the physical layer address of the computer connected to the aforementioned second interface means.

6. Interface device according to Claim 1, characterized in that the aforementioned prescribed format is based on the so-called Ethernet or the so-called IEEE1394 standards.

7. Interface device according to Claim 1, characterized in that the aforementioned second interface means is compatible with multiple types of computer architectures.

8. Receiver characterized in that it is has: a receiver that receives communications from a satellite;

a demultiplexer that that separates the packets for a desired channel from the multiple-channel data streams supplied from the aforementioned receiver;

a depacketizer that reconstructs data from the packets separated by the aforementioned demultiplexer;

a control means that converts the data reconstructed by the aforementioned depacketizer in a prescribed format;

and an interface means that is connected to an external computer.

9. Receiver according to Claim 8, characterized in that the aforementioned control means is controlled by the external computer connected to the aforementioned interface means.

10. Receiver according to Claim 8, characterized in that the data that forms the aforementioned packets is encoded,

and [characterized in that] it is further equipped with a decoding means that decodes the encoded data that has been reconstructed by the aforementioned depacketizer.

11. Receiver according to Claim 8, characterized in that the aforementioned control means is equipped with a correspondence table for Internet protocol addresses and the physical layer address of the computer connected to the aforementioned interface means.

12. Receiver according to Claim 8, characterized in that the aforementioned prescribed format is based on the so-called Ethernet or the so-called IEEE1394 standards.

13. Receiver according to Claim 8, characterized in that the aforementioned second interface means is compatible with multiple types of computer architectures.

Detailed explanation of the invention

[0001]

Technical field of the invention

The present invention pertains to an interface device and a receiver that communicate with the external interface of a personal computer for the purpose of implementing data communication and satellite broadcasting on a personal computer by using a satellite.

[0002]

Prior art

As with receivers for cable television broadcasts and satellite broadcasts, receivers which receive a stream of broadcast audio data and video data and which output the audio/video stream from an external interface are commonly used.

[0003]

In recent years as information systems have become digitized, interest in digital satellite communication technology has grown. Conventional analog satellite broadcasting and satellite communication is primarily used to distribute video, with the distribution of data being an added-

on service, with the systems merely transmitting and receiving the data. However, with the growth of the Internet in recent years the importance of data broadcasting and data communication has begun to be recognized.

[0004]

Consequently, receivers equipped with a function for receiving, in addition to video/audio streams, digital data that is used on a personal computer, are now being offered. Such receivers are capable of outputting a data stream from an output port to a personal computer.

[0005]

Problem to be solved by the invention

However, no receivers have been offered that are equipped with a function for receiving digital data and that also have a function for outputting received broadcast data as an Internet protocol (IP) datagram.

[0006]

Therefore, to use data that has been received via a broadcast line with a personal computer, it is necessary to install a circuit board specific to each [type of] computer, for example, and this offers little versatility and is not necessarily user-friendly.

[0007]

The present invention was devised in light of the aforementioned problems, the objective being to provide an interface device and a receiver with which a satellite broadcast receiver is controlled from a personal computer and with which communication can occur between a device and the personal computer.

[0008]

Means to solve the problem

To solve the aforementioned problem, the interface device according to the present invention has: a first interface means to which is input a data stream received as a communication from a satellite; a demultiplexer that separates the packets for a desired channel from the multiple-channel data streams supplied from the aforementioned first interface means; a depacketizer that reconstructs data from the packets separated by the aforementioned demultiplexer; a control means that converts the data reconstructed by the aforementioned

depacketizer in a prescribed format; and a second interface means that is connected to an external computer.

[0009]

To solve the aforementioned problem, the receiver according to the present invention has: a receiver that receives communications from a satellite; a demultiplexer that that separates the packets for a desired channel from the multiple-channel data streams supplied from the aforementioned receiver; a depacketizer that reconstructs data from the packets separated by the aforementioned demultiplexer; a control means that converts the data reconstructed by the aforementioned depacketizer in a prescribed format; and an interface means that is connected to an external computer.

[0010]

By means of the aforementioned present invention it is possible to communicate with the external interface of a personal computer in order to implement communication and broadcasting on the personal computer using a satellite.

[0011]

Embodiment of the invention

Next, an embodiment of the present invention will be explained with reference to the figures. In the following, a receiver will be explained as a preferred embodiment of the present invention.

[0012]

As shown in Figure 1, the receiver device as an embodiment of the present invention is comprised of a satellite receiver 11 that receives digital broadcasts transmitted from a satellite, and an interface device 20 that processes data from the satellite receiver 11. Interface device 20 is used by being connected to an external personal computer 31.

[0013]

Satellite receiver 11 receives digital signals which are transmitted wirelessly from a satellite, and performs prescribed processing on and outputs [said signals] to interface device 20. Specifically, transmissions from the satellite are received with a front-end device such as a tuner and an MPEG 2 (Moving Picture coding Experts Group Phase 2) transport stream (TS) is extracted. Then, this transport stream is output to interface device 20.

[0014]

Interface device 20 has: an input interface (I/F) unit 21 to which signals from satellite receiver 11 are input; a demultiplexer (DEMUX) 22 that separates packets from the signals from input interface unit 21; a depacketizer (DEPAK) unit 23 that reconstructs signals from the packets separated by separator 22; a decoding unit 24 that decodes signals from depacketizer unit 23; a central processing unit (CPU) 25 that controls each unit of the receiver; and an output interface 26 that outputs signals to external personal computer 31.

[0015]

MPEG2 provides a multi-program handling function enabling multiple programs to be transmitted. This involves the time-division multiplexing of multiple separate encoded streams in relatively short transmission units called transport packets. MPEG2 has two systems for multi-program compatible multiplexing/demultiplexing, these systems being known as a program stream and a transport stream.

[0016]

The header portion of a transport packet has information identifying the packet contents, and based thereupon the packets required for reproduction of the intended program are extracted and decoded via the demultiplexer.

[0017]

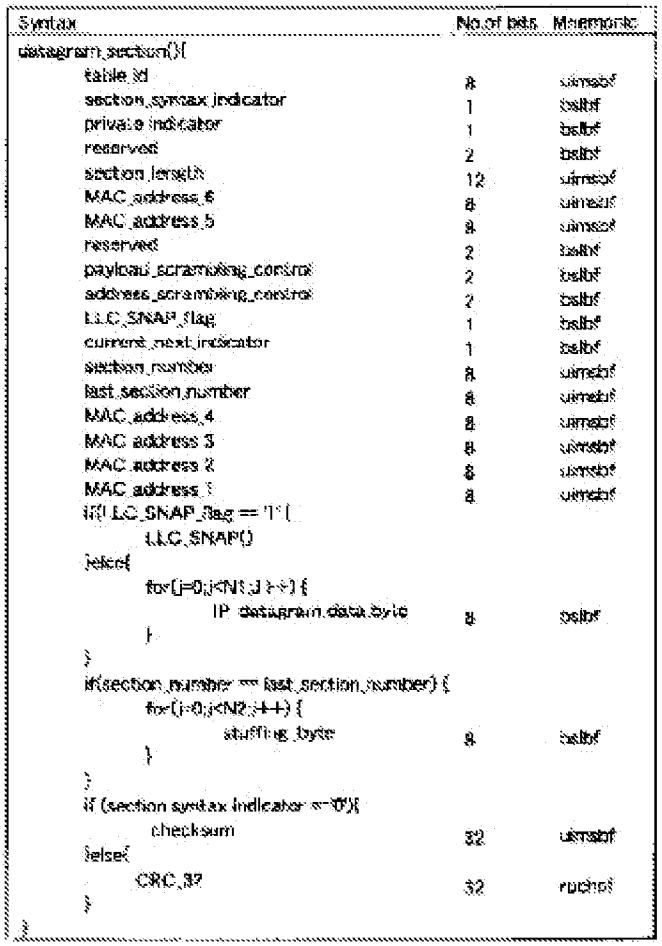
Here, the private section and the internet protocol (IP) datagram of the MPEG2 standard are used as the data format of the upper layer of a transport stream. The transport stream data is divided into IP datagrams.

[0018]

Here, according to the format shown in Table 1, the private section of the transport stream upper layer in the MPEG2 standard encapsulates datagrams. In other words, datagrams are encapsulated within "datagram_sections". The mapping of these "datagram_sections" to an MPEG2 transport stream is defined in the MPEG2 standard.

[0019]

Table 1



Syntax	No. of bits	Mnemonic
datagram_section(){		
table_id	8	uimsbf
section_syntax_indicator	1	bslbf
private_indicator	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
MAC_address_6	8	uimsbf
MAC_address_5	8	uimsbf
reserved	2	bslbf
payload_scrambling_control	2	bslbf
address_scrambling_control	2	bslbf
LLC_SNAP_flag	1	bslbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
MAC_address_4	8	uimsbf
MAC_address_3	8	uimsbf
MAC_address_2	8	uimsbf
MAC_address_1	8	uimsbf
if(LLC_SNAP_flag == '1') {		
LLC_SNAP()		
}else{		
for(j=0;j<N1;j++) {		
IP_datagram_data_byte	8	bslbf
}		
}		
if(section_number == last_section_number) {		
for(j=0;j<N2;j++) {		
stuffing_byte	8	bslbf
}		
}		
if (section_syntax_indicator =='0'){		
checksum	32	uimsbf
}else{		
CRC_32	32	rpchof
}		
}		

[0020]

The configuration of the data of a "datagram_section" is as shown in Table 1.

[0021]

To describe the elements of "datagram sections" in order, [first,] the "table_id" is an 8-bit data field which is set to '0x3E'. The "section_syntax_indicator", "private_indicator", and "section_length" are defined according to ISO/IEC13818-6. "reserved" is a 2-bit field which is set to '11'.

[0022]

"MAC_Address_{1 . . . 6}" is a 48-bit field which includes the destination MAC (Media Access Control) address. The MAC address is divided into six 8-bit fields to which the labels

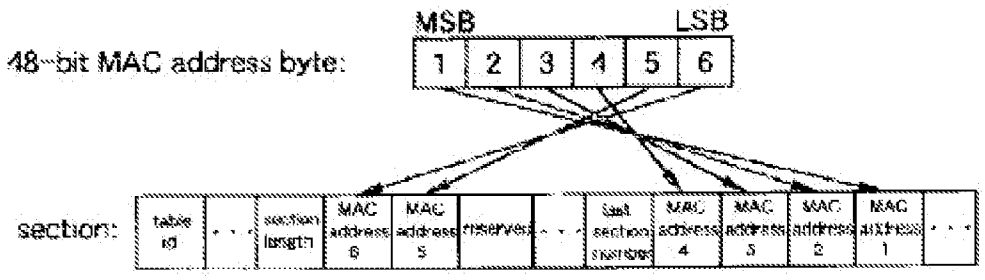
"MAC_address_1" to "MAC_address_6" are appended. "MAC_address_1" includes the uppermost byte of the MAC address; "MAC_address_6" includes the lowermost byte.

[0023]

Table 2 shows the mapping of the MAC address to the section field. The order of the bits in a byte is not an inverse arrangement– the uppermost bit of each byte is moved to the head, unchanged.

[0024]

Table 2



[0025]

As shown in the "address_scrambling_control" field, the "MAC_address" field includes either a distinct [sic; possibly, 'unscrambled'] or an encoded MAC address.

[0026]

As shown in Table 3, "payload_scrambling_control" is a 2-bit field that defines the encoding mode for the section payload. This includes the payload, which begins after "MAC_address_byte_1", but excludes the check sum or the CRC (Circular Redundancy Code) 32 field. The encoding method that is applied is according to the user.

[0027]

Table 3

value	payload scrambling control
00	unscrambled
01	defined by service
10	defined by service
11	defined by service

[0028]

As shown in Figure 4, "address_scrambling_control" is a 2-bit field that defines the encoding mode for this section's MAC address. This field enables dynamic changing of the MAC address. The encoding method is [decided] by the user.

[0029]

Table 4

value	address scrambling control
00	unscrambled
01	defined by service
10	defined by service
11	defined by service

[0030]

The "LLC_SNAP_flag" is a 1-bit flag. When this flag is '1', the payload carries one "LLC/SNAP" encapsulated datagram following "MAC_address_1". The "LLC/SNAP" structure indicates the type of datagram that can be carried. When this flag is '0', the section includes an IP datagram without "LLC/SNAP" encapsulation.

[0031]

"current_next_indicator" is a 1-bit field, the value of which is set to '1'.

[0032]

"section_number" is an 8-bit field. When a datagram is carried in multiple sections, this field indicates the position of the section in the divided process [sic]; otherwise, [the value] is set to zero.

[0033]

"last_section_number" is an 8-bit field, indicating the number of the final section being used to carry the datagram; in other words, it indicates the number of the final section of the division [sic] process.

[0034]

The "LLC_SNAP" structure includes a datagram according to the ISO/IEC8802-2 Logical Link Control (LLC) and the ISO/IEC8802-1a Sub Network Attachment Point (SNAP) specifications. When the section payload is encoded these bytes are encoded.

[0035]

"IP_datafram_data_byte" [sic: "IP_datagram_data_byte"] is the byte that includes datagram data. When the section payload is encoded these bytes are encoded.

[0036]

The "stuffing byte" is an optional 8-bit field, the value of which is not specified in the specifications. When the section payload is encoded these bytes are encoded. In a wide bus environment this supports block encoding and data processing. The number of "stuffing bytes" must be consistent with the requirement for the data alignment defined in the "data_broadcast_descriptor".

[0037]

"checksum" and "CRC_32" are set as defined in ISO/IEC18318-15.

[0038]

Next, the program specification information (PSI) and SI, which pertain to the transmission of the aforementioned data, will be explained.

[0039]

A data broadcast service indicates the transmission of datagrams by including one or more broadcast descriptors. Each descriptor is related to a stream by means of a "component_tag" identifier. In particular the value of the "component_tag" is identical to the "component_tag" field of the "stream_identifier", which exists in a map table for the stream used in transmission of the PSI program datagrams.

[0040]

The "data_broadcast_descriptor" is comprised of the following types of data.

[0041]

"data_broadcast_id" is a field that is set to 0x0005, and is a multiprotocol encapsulation specification.

[0042]

"component_tag" is a field that has the same value as the "component_tag" of the "stream_identifier_descriptor" in the map section of the data stream that is broadcast by the PSI program.

[0043]

"selector_length" is a field that is set to 0x02.

[0044]

The "selector_byte" carries the "multiprotocol_encapsulation_info" shown in Table 5.

[0045]

Table 5

Syntax	No.of bits	Mnemonic
multiprotocol_encapsulation_info () {		
MAC_address_range	3	uimsbf
MAC_IP_mapping_flag	1	bslbf
alignment_indicator	1	bslbf
reserved	3	bslbf
max_sections_per_datagram	8	uimsbf
}		

[0046]

The meaning of the "multiprotocol_encapsulation_info" structure is as indicated in the following.

[0047]

As shown in Table 6, the "MAC_address_range" is a 3-bit field indicating the byte number of the MAC address used by the service to distinguish the receiver.

[0048]

Table 6

MAC address range	valid MAC address bytes
0x00	reserved
0x01	6
0x02	6,5
0x03	6,5,4
0x04	6,5,4,3
0x05	6,5,4,3,2
0x06	6,5,4,3,2,1
0x07	reserved

[0049]

The "MAC_IP_mapping_flag" is a 1-bit flag. When the service uses the IP for the MAC mapping, which is described in RFC1112, this flag is set to '1'. When this flag is '0', the mapping of the IP address to the MAC address is not specified by this specification.

[0050]

As shown in Table 7, the "alignment_indicator" is a 1-bit flag that indicates the alignment between the "datagram_section" bytes and the transport stream bytes.

[0051]

Table 7

value	alignment in bits
00	8(default)
01	32
10	64
11	128

[0052]

"reserved" is a 3-bit field that is set to '111'.

[0053]

"max_sections_per_datagram" is an 8-bit field that indicates the maximum number of sections used to carry a single datagram unit.

[0054]

As for the stream type, the existence of a multiprotocol data stream with the service is indicated by setting the stream type in the program map section of the service to 0x0D or to a user-defined value.

[0055]

MPEG2 provides a multi-program handling function enabling multiple programs to be transmitted. This multi-programming [sic] handling function uses a time-division method to multiplex multiple separate encoded streams in relatively short transmission units called transport packets.

[0056]

MPEG2 has [two] multi-program compatible multiplexing/demultiplexing systems, known as program stream and transport stream. The header portion of a transport packet has information identifying the packet contents, and based thereupon the packets required for reproduction of the intended program are extracted and decoded via the demultiplexer.

[0057]

An MPEG2 transport stream is multiplexed/demultiplexed by means of 188-byte fixed-length transport packets. Because the transport packet itself has a fixed length and is relatively short, the structure is simple; however, because they are streams [comprised] of multiple programs, they are based on hierarchical operational rules [sic].

[0058]

Next, each component that forms receiver 20 will be explained in order.

[0059]

Input interface unit 21 receives a transport stream that is input from satellite receiver 11, and performs prescribed processing with respect to this transport stream. For example, input interface unit 21 is equipped with a FIFO (fist-in first-out) storage means, and it adjusts the speed of the input transport stream and the processing speed of this communication device [sic: receiver] 20.

[0060]

Furthermore, this input interface 21 also inputs and outputs data pertaining [sic; possibly, 'in response to'] to control signals.

[0061]

Demultiplexer 22 separates a prescribed channel from multiple-channel transport streams supplied by input interface 21. In other words, demultiplexer 22 extracts the transport packets with a prescribed packet ID (packet identifier) from the input transport stream.

[0062]

At depacketizer 23 the transport packets of the prescribed channel which have been supplied from demultiplexer 22 are reconstructed as IP datagrams (Internet Protocol datagrams) as defined in the MPEG2 standard private section.

[0063]

When the IP datagrams reconstructed by depacketizer 23 are encoded, decoding unit 24 performs a decoding process.

[0064]

Here, the IP (Internet Protocol) is a connectionless protocol which controls the transmission and reception of data packets called IP datagrams within networks and between networks without continuing to maintain a connection during communication.

[0065]

The IP has functions such as: the setting and identification of the network – that is, the IP addresses that are the Internet addresses; processing of IP datagrams; and control of the communication path until the IP datagrams arrive at their destination [literally, 'the other party'].

[0066]

In other words, the IP is the TCP (Transmission Control Protocol)/IP network layer protocol, delivering packets within the Internet.

[0067]

Specifically, the IP uses an IP address such as a computer ID number at [sic; possibly, 'as'] the address used in the network based on the TCP/IP protocol, and uses IP multicast, which is a communication technology whereby a single packet is delivered to multiple users simultaneously.

[0068]

The datagrams processed by decoding unit 24 are processed by CPU 25 according to the medium of [sic; possibly, 'associated with'] output interface unit 26. For example, when output interface 26 is a so-called Ethernet controller, an Ethernet frame is constructed and passed to output interface 26. Furthermore, for example, when output interface 26 is an IEEE1394 link chip, an IEEE1394 frame is constructed and passed to output interface 26.

[0069]

The data that has been processed by CPU 25 according to the media type is output externally by output interface 26 in accordance with a data link protocol. For example, output interface 26 outputs the data supplied from CPU 25 in accordance with the data link protocol for the Ethernet or for IEEE1394.

[0070]

Furthermore, this output interface 26 also inputs and outputs data pertaining to control signals.

[0071]

Personal computer 31 receives data transmitted from interface device 20. In other words, personal computer 31 receives the so-called Ethernet frames or the so-called IEEE1394 frames, for example, transmitted from output interface 26 of interface 20, as the computer-side interface.

[0072]

Here, the transmission destination address for the data link used when data is output is required. When the IP datagram is unicast, interface device 20 uses a data link address resolution protocol (ARP) to recognize the physical layer address of the transmission source personal computer 31.

[0073]

Furthermore, if the IP datagram is multicast, the interface device uses a multicast address. The multicast method differs according to the physical network. For example, with Ethernet the packets within a segment run on the network, so implementation is easily accomplished.

[0074]

The receiving node extracts packets having, as the destination, [the receiving node's] own Media Access Control (MAC) address, a broadcast MAC address, or – when it belongs to a multicast – the MAC address corresponding to that group.

[0075]

Next, the aforementioned ARP will be explained together with the reverse address resolution protocol (RARP), which is the reverse process of the ARP.

[0076]

ARP/RARP are protocols that automatically mutually exchange a 32-bit address and an Ethernet address, which is a 48-bit physical address, and the MAC address; [these protocols] are positioned between the network layer and the IP layer.

[0077]

Thus a system at a layer higher than the IP layer only needs to know the IP address. Furthermore, as for the physical address, an Ethernet board-specific physical address is statically assigned by the [device] manufacturer, so the IP address can be flexibly created/changed by means of software.

[0078]

The ARP broadcasts an ARP request message – which designates the other party's IP address, and is used to determine the physical address for the other party, whose IP address [already] is known – to the entire system. The system corresponding to the designated IP address knows its own physical address and IP address, so it sends to the original requesting system a reply message in which that physical address and IP address have been combined. Based on this reply, the original system creates/updates an entry for the combination of that physical address and IP address.

[0079]

The RARP, which is the reverse of the ARP, is used by a system that knows its own physical address to determine the IP address for the [larger] system in which it is included; it sets the physical address of the system whose IP address it wants to know, and broadcasts a RARP request message – in other words, an IP address request – with that system as the recipient.

[0080]

A RARP service, which responds to the request, occurs on the network; in other words, there is one lowest RARP server capable of replying to the RARP request message, and that system replies directly to the system from which the request originated with the required information, which is the combination of the physical address and the IP address.

[0081]

The messages used with ARP/RARP have the same format, and are identified based on the protocol type. The other fields contain the Ethernet and similar hardware interfaces, the ARP request/reply, the RARP request/reply, and similar operations, and the physical address/IP address being transmitted, for example.

[0082]

The ARP/RARP will be explained in more detail hereinafter.

[0083]

A broadcast, which translates [into Japanese] as 'broadcast', is the transmission of the same information to all of the nodes in a given system. In contrast thereto, a 1-1 communication is called a 'unicast', and a multicast is [conceptually] between [a broadcast and a unicast].

[0084]

When Ethernet is used, and if it is a multicast, the interface device transmits with an address that combines the lower 25 bits of the IP transmission origin address and the lower 23 bits of '01:00:1E:00:00:00', as stipulated in RFC1112.

[0085]

Next, the [format of the] data at each stage in the receiver will be explained with reference to Figure 2.

[0086]

As shown in A in Figure 2, the signal 'a' transmitted to input interface 21 of interface device 20 from satellite receiver 11 in Figure 1 is a packet in which the Transport Stream (TS) header is followed by the Internet Protocol (IP) header, after which is the data. Alternatively, as shown in B in Figure 2, it is a packet in which the data directly follows the TS header.

[0087]

As shown in C in Figure 2, the signal 'b' transmitted to demultiplexer 22 from input interface 21 of interface device 20 is a packet in which the Transport Stream (TS) header is followed by the Internet Protocol (IP) header, after which is the data. Alternatively, as shown in D in Figure 2, it is a packet in which the data directly follows the TS header.

[0088]

As shown in E and F in Figure 2, the signal 'c' transmitted to depacketizer 23 from demultiplexer 22 is one for which the TS header has been removed from the head of a packet having the data configuration shown in C and D in Figure 2. The demultiplexer separates the signal in accordance with the TS header and then removes the TS header.

[0089]

As shown in G in Figure 2, the signal 'd' transmitted from depacketizer 23 to decoding unit 24 has the encoded data following the IP header. Depacketizer 23 decodes this encoded data and, as shown in H in Figure 2, transmits a packet in which the decoded data follows the IP header to CPU 25.

[0090]

CP 25 adds, to the signal 'e' transmitted from depacketizer [sic decoding unit] 24, a data link header pertaining to the external medium, and transmits [the resulting signal] to output interface 26.

[0091]

Output interface 26 outputs the signal 'f' – transmitted from CPU 25 – to external personal computer 31 as the signal 'g'. Signal 'g' and signal 'g' [sic], as shown respectively in I and J in Figure 2, append the IP header and the data link header to the decoded (raw) data.

[0092]

Next, the method whereby the aforementioned receiver is controlled from external personal computer 31 will be explained.

[0093]

To control this receiver from personal computer 31, basically UDP (User Datagram Protocol)/IP is used.

[0094]

UDP is a protocol that is known as a connectionless protocol; it does not perform flow control or sequence control, nor does it establish a connection, for example.

[0095]

As shown in Figure 3, the basic format for UDP/IP is comprised of an IP header, a UDP header, an operation code, and operation data.

[0096]

The operation codes for a control packet from the personal computer with respect to satellite receiver 11 of the receiver are as shown in Table 8.

[0097]

Table 8

Operation Code	データバイト数 (A)	(B) 意味
0x0001	2byte	PowerのON/FF 0:OFF 1:ON
0x0002	2byte	SIDの設定 (C)
0x0003	2byte	Componentの設定 (C)
0x0004	2byte	PIDの設定 (C)
0x0005	2byte	Local周波数の設定 (D)
0x0006	2byte	偏波の設定 (E)

Key: A Data byte count
B Meaning
C Setting
D Frequency setting
E Polarization Setting

[0098]

In summary, operation code "0x0001" pertains to power control; "0" signifies power off, and "1" signifies power on. Operation code "0x0002" signifies the SID (Stream Identification) setting. Operation code "0x0003" signifies the component setting. Operation code "0x0004" signifies the PID (Packet Identifier) setting. Operation code "0x0005" signifies the local frequency setting. Operation code "0x0006" signifies the polarization setting.

[0099]

All of the aforementioned control packets for the receiver have a data byte count of two bytes. The control contents are transmitted as the operation data which follows this operation code.

[0100]

Here, the PID is 13-bit stream identification information indicating the packet's individual stream. Furthermore, '0x' signifies a hexadecimal representation.

[0101]

The personal computer constructs and outputs to interface device 20 a UDP datagram that includes an operation code of this type, and thus controls satellite receiver 11. CPU 25 of interface device 20 interprets the UDP datagram supplied from personal computer 31, converts it to the protocol for satellite receiver 11, and controls satellite receiver 11.

[0102]

In addition, the internal parameters of the receiver must be controlled. In this case too, as with the aforementioned control of satellite receiver 11, a format that is based on a UDP datagram is used.

[0103]

In other words, as shown in Table 9, with a control packet for the receiver the operation code "0x0101" pertains to power control, with "0" signifying power off and "1" signifying power on. For operation code "0x0102", "0"signifies Ethernet selection and "1" signifies IEEE1394 selection. With these control packets for the interface device, the data byte count is also two bytes.

[0104]

Table 9

Operation Code	データバイト数 (A)	(B) 意味
0x0101	2byte	Power ON/OFF 0:OFF 1:ON
0x0102	2byte	メディアの選択 0:Ethernet 1:IEEE1394 (C)

Key: A Data byte count

B Meaning
C Media selection

[0105]

Next, the aforementioned ARP/RARP will be explained in detail.

[0106]

The address resolution protocol (ARP) provides a resolution by means of dynamic linking, so a new machine can be added without recompiling code. Furthermore, an efficient and easily managed structure is maintained without having to maintain a central database. With ARP, the use of a lower-level protocol to dynamically link addresses has been selected, so that a correspondence table does not have to be maintained.

[0107]

As shown in A in Figure 4, with the dynamic linking used by ARP, when a host A attempts to resolve an IP address IB, a special packet is broadcast, requesting that the hosts of IP address IB reply with physical address PB. As shown in B in Figure 4, all of the hosts including B receive the request, but only host B recognizes that IP address and returns a reply which includes the physical address. A knows the physical hardware address reply [sic; possibly, 'based on that reply'] of B, and uses that address to directly transmit internet packets to B.

[0108]

Thus, with ARP a given host is able to discover the physical address of another host on the same physical network merely by supplying the other party's IP address.

[0109]

As for the relationship between ARP and other protocols, ARP is one possible mechanism for associating an IP address with a physical address. Many network technologies have come to see that this is unnecessary. What is important is the ARP will be completely unnecessary if all of the network hardware can be enabled to recognize an IP address. Accordingly, regardless of the address structure at the lower level that is used by the hardware, the ARP only adds a new address structure thereupon.

[0110]

Thus, ARP is a lower-level protocol that hides the physical addressing of a basic network, enabling IP addresses to be allocated freely to any machine. ARP can be thought of as one part of the physical network system, and not as one part of the Internet protocol.

[0111]

As for ARP encapsulation and identification, when ARP messages are transmitted from a given machine to another machine, they must be included in and carried in the physical frame. Figure 5 illustrates the inclusion and carrying of an ARP message in the data portion of a frame.

[0112]

To identify a frame carrying an ARP message, the transmitting party assigns a special value to a type field in the fame header, and the ARP message itself is contained in the frame's data field. When the frame arrives at a machine, the network software uses the frame type to determine what is contained therein. Almost all technology uses a single type value with respect to all frames that carry an ARP [message]. The recipient's network software must additionally check the ARP message to distinguish whether it is an ARP or an address resolution protocol reply implementation [sic].

[0113]

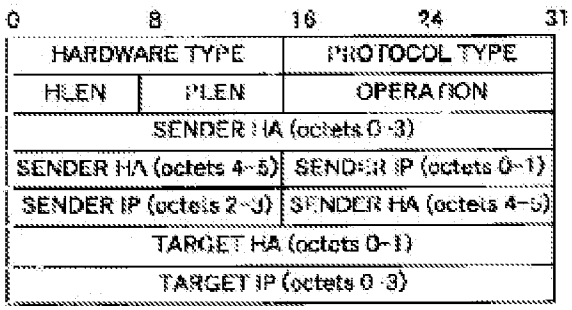
The format of an ARP differs from that of most protocols; the ARP packet's header does not have a fixed format. Instead, the length of the field containing the address depends upon the network type, so that the ARP can be used by various network technologies. However, the header contains, near the beginning, a fixed field that specifies the length of the address contained in the subsequent field, so that any ARP can be interpreted.

[0114]

The actual ARP message format has a format that is general enough to be able to be used with any physical address and any protocol address. The example in Table 10 shows a 28-octet ARP message format that can be used with Ethernet hardware when a 4-octet long IP protocol address is being resolved. Here, the physical address is 48 bits – six octets – long.

[0115]

Table 10



[0116]

[This is] an example of an ARPR [sic; RARP]/ARP message format used to resolve an Ethernet address from an IP address. The length of the field [or: 'fields'] depends on the length of the hardware address and the protocol address, with six octets for the Ethernet address and four octets for the IP address.

[0117]

The RARP is a separate protocol that uses the same message format.

[0118]

Table 10 shows the standard format used throughout this book [sic], with the ARP being expressed with four octets per one line. Unfortunately, unlike most other protocols, the variable-length fields in the ARP packet do not neatly conform to a 32-bit limit, making it difficult to read the diagram. For example, the transmitting party's hardware address, which is titled "SENDER HA" occupies six consecutive octets, but in the diagram it stretches across two lines.

[0119]

The "HARDWARE TYPE" field shows the hardware interface type for which the transmitting party is seeking a reply. This is a value [of] 1 with respect to Ethernet. Similarly, the "PROTOCOL TYPE" field specifies the value for the upper level protocol address supplied by the transmitting party. This includes 0800_{16} with respect to the IP address. The "OPERATION" field indicates an ARP request (1), an ARP reply (2), a RARP request (3), [or] a RARP reply (4). The "HLEN" field and the "PLEN" field indicate the physical hardware address length and the upper [level] protocol address length, with the ARP being capable of being used with any network. If the transmitting party knows that hardware address and IP address, these are provided in the "SENDER HA" field and the "SENDER IP" field, respectively.

[0120]

When a request is created, the transmitting party provides the target IP address (ARP) or the hardware address (RARP) in the "TARGET HA" field and the "TARGET IP" fields. Before the machine target machine replies, the address is embedded during that interval, the "TARGET" and "SENDER" are switched in the combination [thereof], and "OPERATION" is changed to 'reply'. Accordingly, the reply carries the IP address and hardware address for the machine [literally, 'object'] which originated the request, together with the IP address and hardware address for the machine for which a linkage is being requested.

[0121]

The IP address is assigned independently of the machine's physical hardware address. To transmit an Internet packet from a given computer to another computer though a physical net[work], the network software must associate the IP address with the physical hardware address and transmit the frames using the hardware address. When there are fewer hardware addresses than IP addresses, the machine's physical address can be directly mapped by encoding it within the IP address. When that is not the case, the association must be performed dynamically. The ARP resolves a dynamic address using only the lower level network communication system. Thus, the machine is able to resolve the address without maintaining a persistent record of the address associations.

[0122]

A machine uses ARP to broadcast an ARP request in order to discover another machine's hardware address. That request includes the IP [address] of the machine whose hardware address is needed. All of the machines on the network receive the ARP request. When the [address in the] request matches a machine's IP address, that machine transmits a reply which includes the needed hardware address. The reply is sent to one machine; it is not broadcast.

[0123]

To make the ARP efficient, each machine caches the link from the IP address to the physical address. There is a significant tendency for Internet traffic to be comprised of a series of exchanges between specific machines, and the cache reduces a large portion of ARP broadcasts.

[0124]

As explained above, the present embodiment pertains to a communication device for the purpose of connecting a satellite broadcast receiver and a computer, with a focus on digital

satellite broadcasting and Internet technology; specifically, it pertains to a communication protocol for the purpose of controlling a satellite broadcast receiver from a computer and performing communication between a main device and a computer.

[0125]

In other words, the present embodiment is an interface device for the purpose of receiving data output from a satellite receiver in satellite broadcasting using a satellite line or in satellite communication, and performing a conversion to the protocol for the external interface of a personal computer, with communication being performed between this interface device and the personal computer, and IP datagrams or the MPEG2 private section level being decoded [sic]. Furthermore, [the present embodiment] pertains to a receiver comprised of the aforementioned interface device and a satellite receiver.

[0126]

Furthermore, with the present embodiment, a personal computer was used as the example of the other party that outputs datagrams externally, but the present invention is not limited thereto. For example, [the interface device] can be connected to a server that records/reproduces datagrams.

[0127]

Effect of the Invention

As described above, by means of the present invention it is possible to connect to a versatile interface of a computer irrespective of the interface standard for a satellite receiver.

[0128]

Furthermore, because the restrictions to the computer-side interface are reduced, it can be used by machines with many architectures, and many operating systems (OS). For example, it is compatible with differences in architectures and OS's such as [those for] workstations, personal computers, and notebook personal computers.

[0129]

Furthermore, the present invention can be applied to communication devices that perform bidirectional communication and to capability-expanding devices for broadcast systems, and, for example, [it can be applied to] the saving of data, recording to a so-called MD, and recording to a so-called DVD.

Brief description of the figures

Figure 1 is a schematic block diagram showing the configuration of the receiver.

Figure 2 is a diagram showing the configuration of the packet data for each part of the receiver.

Figure 3 is a diagram showing the basic format of a control packet.

Figure 4 is a diagram explaining dynamic ARP resolution.

Figure 5 is a diagram explaining ARP encapsulation.

Explanation of symbols

11 Satellite receiver
20 Interface device
21 Input interface unit
22 Separator
23 Depacketizer
24 Decoding unit
25 CPU
26 Output interface unit
31 Personal computer

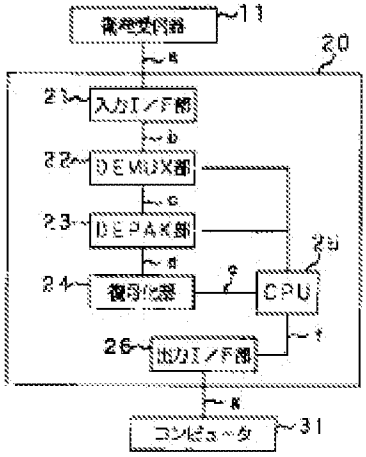


Figure 1. Functional block diagram

Key: 11 Satellite receiver
21 Input I/F unit
22 Demux unit
23 Depak unit
24 Decoding unit
26 Output I/F unit
31 Computer

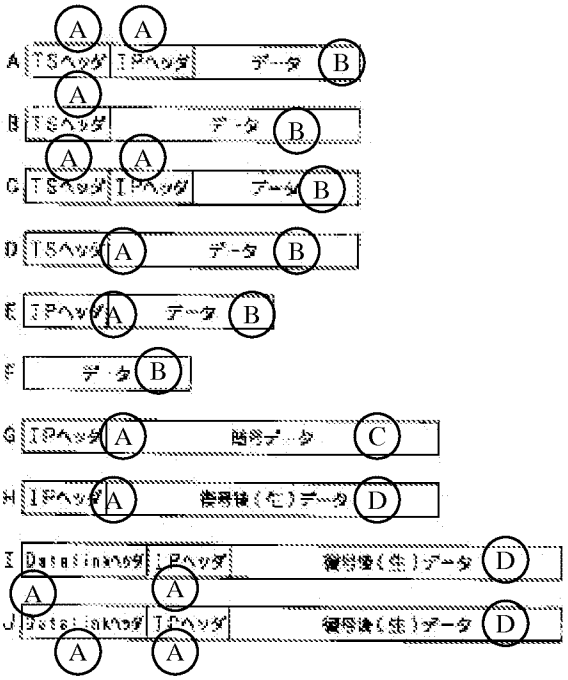


Figure 2

Key: A Header
B Data
C Encoded data
D Decoded (raw) data

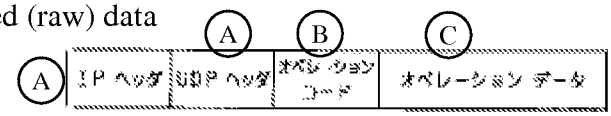


Figure 3. Basic format of control packet

Key: A Header
B Operation code
C Operation data

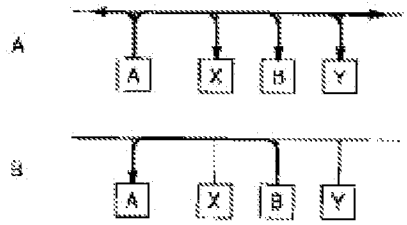


Figure 4. ARP protocol

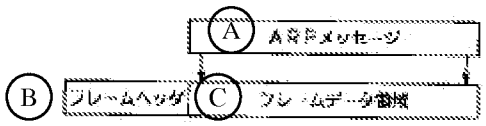


Figure 5. Encapsulated ARP message inside physical network frame

Key: A ARP message
B Frame header
C Frame data region